Human Factors in Cybersecurity, Vol. 168, 2025, 29–39
https://doi.org/10.54941/ahfe1006140

Leveraging Complex Access Scenarios (CAS) to Bridge Human-Centered HCI

Rahmira Rufus

AWT Solutions, LLC, ARTEK Academy, Silver Spring, MD 20910, USA

ABSTRACT

Within the realm of human-computer interaction (HCI), the shift from traditional stimulus-response models to more integrated human-computer partnerships mark a significant technological evolution, which is coined as human-centered AI (HCAI). This shift is driven by the advent of autonomous agents and AI, which transform HCI from simple interactions to complex integrations where systems anticipate user needs and collaborate effectively. This integration challenges us to design systems that are not only efficient and safe but also intuitive, aligning closely with human behavior and expectations. Addressing these challenges brings about the opportunity to focus on technical objectives that are crucial in shaping the future of HCI to effectively incorporate HCAI. In this work, Complex Access Scenarios (CAS) are leveraged to not only reveal system complexity but also to propose a method to bridge HCI-to-HCAI as 'Human-Centered HCI'.

Keywords: Human-centered HCI, Complex access scenarios, Complex computing systems, Artificial intelligence, Human-computer interaction, Autonomous agent, Human-computer integration, Human-centered AI, Context awareness, Situational awareness, Cybersecurity

INTRODUCTION

Recent technology innovation has been transforming numerous industries, reshaping how we interact with the digital world. Some key next-generation technological advancements include AI/ML, blockchain, quantum computing, 5^{th}-6^{th} generation technology and Internet of Things (IoT). Amongst these innovations emerged a concept called Complex Access Scenarios (CAS). CAS is an access event-modeling scheme focused on situations where users or systems encounter intricate access control requirements, necessitating advanced authentication and authorization mechanisms (Rufus, 2021). Specifically, CAS addresses the service concerns arising for the nuanced demands of modern computing systems. CAS emphasizes contextual and situational awareness, which is crucial in environments involving human-computer interaction (HCI) and human-centered AI (HCAI), making complex systems more intuitive and user-friendly (Rufus, 2023).

AI/ML has revolutionized industries by enabling automation, predictive analytics and enhanced decision-making. However, the autonomy, autonomic and autonomous capabilities that AI/ML offers form the basis for the 'perception dilemma' discovered in a 2003 HCI investigation by

Anderson et al. Even though HCI has since progressed from the 20-year-old investigation, the fundamental HCI premise is still based upon the "users act, and systems react" condition (Anderson et al., 2003). The earlier CAS work Rufus conducted focused on this perception dilemma in combination with an interaction-to-integration leap discovered by Farooq and Grudin to determine if CAS was applicable to the HCI and HCAI space (Farooq and Grudin, 2016). The conclusion was that there exists a nexus amongst the two key issues analyzed, and this paper continues where the previous analysis concluded. The claim was that CAS is applicable to the HCI-to-HCAI issue, however the research conducted in that body of work only determined whether CAS criteria could be established. In this work, we identify some technical objectives and challenges in this space to demonstrate why this approach is crucial in developing systems that adapt to the evolving landscape of data services and networking technologies, enhancing accessibility and user experience.

LEVERAGING COMPLEX ACCESS SCENARIOS (CAS)

The basis for establishing CAS criteria is to develop an event-modeling framework that: (1) ensures classification based on complexity factors such as properties, characteristics, and conditions; (2) links access criteria to the appropriate "complexity" classification or otherwise reclassifies it as "complicated"; and (3) verifies that the request is system-generated. By examining context and situational awareness in HCI and assessing complexities associated with system resource access, it was determined that CAS did exist (Rufus, 2023). To specifically examine usability factors in the HCI scenario that determined whether CAS criteria was applicable, this work revisits the interaction-to-integration example as a use case (see Figure 1). This is where the research explores the complexities of interconnected data and web environments, which were identified but not quantified in the initial CAS service request studies. At this stage, we aimed to find simple correlations that are prone to misrepresentation and illustrate them through a working CAS model, because such misrepresentations serve as key elements in verifying the presence of CAS. The verification is conducted via the CAS framework, where CAS criteria requirements and specifications are emphasized.

Complex Access Scenarios (CAS) Framework

The CAS framework is constructed to establish that complexity is confirmed via events that satisfy required criteria specified for a given domain (see Table 1). In previous CAS work and related work that referenced CAS, only CAS requirements were utilized. In this work, CAS requirements and specifications are further explained in detail to demonstrate how the evaluation of an access event is dissected within the CAS framework. For this paper, leveraging CAS for the HCI domain revealed not only the complexity of access requests within this space, but also the opportunity to utilize this output as a basis (as input) to develop a counter-methodology to bridge the concern of the HCI progression. The concern is the leap from the manual

execution of system tasks that jump from HCI to HCAI without effectively evaluating if the transition is as smooth as marketed (confirmed AI outputs against manual checks of expected outcomes). Therefore, usage scenarios were developed to uncover issues within the HCI space. While the CAS framework was being leveraged to conduct this investigation, a strategy emerged that could bridge the gaps within the HCI-to-HCAI to derive the notion of a Human-Centered HCI as the intermediary. In the next section, an explanation of the CAS framework is provided, along with the use cases explored to establish the groundwork for the CAS declarations. Following the execution of the framework are the early findings for our proposed Human-Centered HCI methodology.

Table 1: CAS framework criteria.

Requirements	Specifications	
1	Complexity Classification	**Properties**: intrinsic attributes of a system or entity that define its nature, which remain constant regardless of context. **Characteristics**: observable traits or behaviors emerging based on system interactions that may change depending on external factors. **Conditions**: refers to situational factors or constraints impacting how a system or entity functions; defines the context in which properties and characteristics operate.
2	Complex or Reclassify (cannot be downgraded)	**Complex**: involves unpredictability, interconnectivity, and real-time adaptation based on external and internal conditions. **Complicated**: not easy to understand, execute or perform but able to accomplish after employing predicable, structured problem-solving processes. **Simplistic**: easy to understand, execute or perform.
3	System Generated Request (not user-initiated)	**System Generated**: occurs when a computing system autonomously initiates an access request based on internal triggers, environmental conditions, or predictive analytics. This occurs independent of direct user input, otherwise the request is user-initiated.

1st CAS Framework Criteria: Complexity Classification

To establish CAS based on the first criterion— complexity classification— we need to differentiate between properties, characteristics and conditions via an HCI/HCAI based example and illustrate how these specifications contribute to the complexity assessment. Based upon the criteria prescribed for CAS classification in Table 1, a system or scenario must exhibit a combination of properties, characteristics, and conditions that contribute to complexity. Table 2 provides examples for system properties, characteristics and conditions commonly known for computing environments.

Table 2: CAS complexity classification criteria examples.

CAS Classification	Classification Specification Example
Properties	System's encryption protocol (e.g., AES-256, Triple-DES) is a **property** example, because it defines a fundamental aspect of security.
Characteristics	Network latency is a system **characteristic** because it varies based on traffic load, bandwidth and server response time.
Conditions	Firewall rules allowing SSH access is a **condition** that dictates access permissions based on security policies.

Table 3 provides an HCAI CAS Classification example for an AI-driven access control (AC) system. This classification helps determine whether an access scenario is complex or should be downgraded to complicated based on its dynamic, adaptive, and multi-layered nature.

Table 3: HCAI CAS classification example (AI-driven access control (AC) system).

CAS Classification	AI-Driven Access Control (AC) System Example
Properties	Uses a multi-factor authentication (MFA) system (biometric, OTP, password).
Characteristics	Adaptive authentication modifies access levels based on user behavior and risk factors.
Conditions	If an unusual login attempt occurs from a new device and location, additional verification steps (such as manual admin approval) may be required.

Next, we illustrate properties, characteristics, and conditions for an HCI scenario with the AI-driven AC system for CAS complexity classification. The AI-driven access control is in a collaborative work environment for a virtual workspace. Company ABC implements the AI-powered AC system to regulate entry to its virtual workspace, where users collaborate on sensitive projects. The system dynamically adjusts access permissions based on real-time behavior, user roles, and other contextual factors. Some complexity classification criteria for the AI-driven AC system are the following:

- Properties (Intrinsic System Attributes)
  - AI-driven authentication: Uses biometric verification, behavioral analysis, and role-based access.
  - Encryption protocols: Protects data during transmission and storage.
  - Access logging mechanisms: Maintains an immutable log of user actions.
- Characteristics (Dynamic and Contextual Traits)
  - Adaptive user authentication: Adjusts security measures based on login patterns and risk analysis.
  - User behavior monitoring: Detects anomalies (e.g., a sudden login from an unrecognized country).

  - Access prediction modeling: Anticipates access needs based on work patterns.

- Conditions (Situational Constraints and Policies)
  - Conditional access rules: If a user logs in from a high-risk location, an additional security verification step (e.g., human administrator approval) is triggered.
  - Time-based restrictions: Certain files can only be accessed during designated work hours.
  - Multi-user collaboration constraints: Access is granted differently based on team roles and ongoing project security levels.

Each classifier contributes to complexity because the properties are inherent to the AC system and contribute to its baseline security complexity, which is necessary but not necessarily sufficient to classify it as a CAS alone. The characteristics emerge based on user interaction with the system. Since they fluctuate, they introduce variability that makes access control more complex. These conditions introduce dynamic, situational dependencies that affect access, requiring real-time decision-making by both the system and the user.

To satisfy this CAS class, the system must demonstrate: (A) combination of properties, characteristics, and conditions that interact dynamically; (B) autonomous decision-making (e.g., adaptive authentication); and (C) context-driven security responses (e.g., behavior-based access restrictions). The AI-driven AC system meets the initial CAS criteria because:

- It adapts autonomously based on evolving scenarios.
- It integrates security policies with real-time contextual awareness.
- It introduces multi-layered complexity that cannot be reduced to a "complicated" system.

Some key takeaways to utilize for establishing CAS classification criteria from this HCI example:

- properties define system capabilities but do not alone establish complexity.
- characteristics introduce variability that influences decision-making.
- conditions impose dynamic constraints, making access control adaptive and non-linear.

2nd CAS Framework Criteria: Complexity Sustainment

The 2nd CAS criteria to establish builds on the last bullet established in the AI-driven AC system. After complexity is introduced, can it be reduced to complicated? In the context of HCI evolving into HCAI, distinguishing between a complex, complicated and simplistic system structure is crucial for establishing this criterion. However, to progress being assessed for the 3rd criteria, concatenating access criteria to the complex class must be sustained and cannot be downgraded to complicated. The reason is demonstrated in the following simple, complicated and complex system comparisons via Table 4.

Table 4: CAS 2nd criteria: System comparisons examples.

CAS Type	Criteria	Example
Simple	• Operate on basic rules and predictable patterns with little to no dynamic adaptation. • Don't require real-time contextual awareness or sophisticated decision-making.	Static password-based login system where users enter credentials without any adaptive security measures.
Complicated	Consists of multiple interdependent components but follows predictable rules and can be broken down into structured, step-by-step processes for problem-solving.	• MFA system with fingerprint scanning, OTP verification, and security questions. • While it adds layers of security, it still follows fixed procedures and doesn't autonomously adjust based on user behavior.
Complex	• Involve unpredictability, inter-connectivity and real-time adaptation based on external and internal conditions. • Integrate ML/AI-driven decision-making and continuous feedback loops.	Our AI-powered AC that continuously learns from user behavior, location and device risk to dynamically adjust access permissions.

Now that the criteria are established via Table 4, for a system to remain classified as complex under CAS, its access mechanisms must be:

- Non-linear - decisions are based on evolving patterns.
- Context-aware - analyzes behavioral, environmental and system variables.
- Adaptive - modifies access dynamically based on risk assessment.

Therefore, with the AC system as a security control system, if the system continuously evolves, self-adjusts, and predicts access risks dynamically, it remains classified as complex under CAS. However, if it is discovered that the system for some reason can apply predefined static rules, it did not sustain the 2nd CAS complexity requirements even though in the 1st classification criteria system complexity was introduced. The system should be revaluated and classified as complicated rather than complex. An example use case was developed to establish the 2nd CAS criterion in our HCI to HCAI pathway by modeling the transition from HCI to HCAI for ABC's AI-driven AC to demonstrate how secure data access would occur. The model uses the AC system's access scenarios and categorizes via Table 5 how the system would execute each as simplistic, complicated or complex processes (access criteria).

Table 5: HCI-to-HCAI progression for 2nd CAS criteria example.

Scenario	Simple	Complicated	Complex
Login Authentication	Password-based system	MFA with fixed steps (password-OTP-biometric)	AI-driven adaptive AC that adjusts security dynamically
User Behavior Analysis	No behavior tracking	Basic Rule-based monitoring (flagging login attempts from unknown locations)	Continuous AI-driven behavioral analysis that predicts risks and adjusts permissions accordingly
Threat Detection	No detection	Predefined security alerts	AI autonomously detects anomalies and blocks unauthorized access in real-time
Access Modification	Static permissions	Role-based access control (RBAC)	Adaptive permissions that change based on user context, workload, and risk profile

Some challenges that arose in establishing the 2nd CAS criterion in this HCI to HCAI use case example:

- Computational Overhead - AI-driven adaptive systems require high processing power and real-time computation, making implementation costly.
- Security Risks in AI Autonomy - An AI system that modifies access dynamically could be exploited if adversaries manipulate its learning models.
- Bias and Ethical Concerns - ML models may discriminate against certain users due to biased training data.
- Explainability and Transparency - Unlike rule-based systems, AI decisions can be opaque, making it difficult to audit access decisions.
- Scalability - Implementing AI-driven CAS across large enterprises with multiple access layers is logistically complex. Being able to reduce this classification to complicated is highly unlikely and will often guarantee a complex classification, which are viable research usage scenarios.

3rd CAS Framework Criteria: System Generation

The 3rd criteria in establishing a CAS determination, is if the requests are system generated and not user initiated. To continue with the use case example for Company ABC's AC system, some examples can be the following:

- AI-driven privilege escalation - A ML system detects an administrator's increased workload and proactively grants temporary elevated access to necessary system resources.
- Automated security authentication - A security system detects an anomaly (e.g., a sudden login from a new location) and requests additional authentication measures dynamically.

- Predictive resource allocation - A cloud-based system anticipates a spike in computational demand and automatically requests additional CPU/GPU resources.

To make such a determination, a request is classified as system-generated when it meets the following criteria illustrated via Table 6. Examples are also provided in addition to the definition for each system-generated criteria element.

Table 6: CAS 3rd criteria: System-generated request determination.

Criteria	Description	Examples
Autonomy	Initiated by the system without human intervention.	AI dynamically increases security verification for a flagged user session.
Event-Driven	Triggered by predefined conditions, thresholds, or anomalies.	System detects unusual access behavior and restricts permissions.
Predictive Analytics	ML anticipates resource needs and preemptively requests access.	AI predicts an increased user workload and grants temporary admin rights.
Policy-Based	Follows a set of predefined rules and compliance policies.	System enforces zero-trust security by requesting just-in-time access verification.

System-generated determinations for ABC's AC system utilized the following access scenarios displayed in Table 7. Therefore, within the scope of this HCI-to-HCAI use case example, if access requests are autonomously triggered by AI/ML or event-driven policies, they qualify as system-generated and remain in the complex category. Otherwise, if a request is only rule-based without autonomous adaptability, it is classified as complicated rather than complex.

Table 7: HCI-to-HCAI progression for 3rd CAS criteria example.

Access Scenario	User-Initiated Request	System-Generated Request
Security Clearance Adjustment	A user manually requests elevated privileges.	AI detects an urgent need and auto-grants temporary privileges based on behavioral analysis.
Resource Scaling	An administrator increases cloud storage manually.	System predicts a storage surge and automatically requests additional space.
Threat Mitigation	A security analyst blocks a suspicious login manually.	AI identifies a threat pattern and auto-restricts access.
Device Access	A user requests VPN access from a new location.	The system detects an unverified device and triggers additional authentication.

Some challenges that arose in establishing the 3rd CAS criterion in this HCI to HCAI use case example:

- Detection Performance Metrics Discrepancies w/ False Positives & False Negatives - AI systems may mistakenly grant or deny access due to misclassification.
- Computational Complexity - Predictive analytics requires high processing power, making real-time decisions difficult at scale.
- Ethical & Privacy Concerns - Autonomous access control may overstep user consent and introduce surveillance risks.
- Vulnerability to Adversarial Attacks - Hackers could manipulate AI decision-making through data poisoning or adversarial learning to gain unauthorized access.
- Policy Compliance & Regulation - AI-driven AC must comply with cybersecurity laws (e.g., GDPR, NIST Zero Trust Architecture).

BRIDGING HCI-TO-HCAI WITH HUMAN-CENTERED HCI

The transition from HCI to HCAI signifies a pivotal shift toward designing AI systems that prioritize human needs, values, and ethical considerations. This progression is guided by Human-Centered HCI, a framework focused on integrating AI technologies while maintaining human agency and enhancing user experience. With Human-Centered HCI being an intermediary to bridge HCI and HCAI effectively, we can address misrepresentations that occurred in the previous issues explored.

Interaction-to-Integration Usability Misrepresentation

As HCI transitions from its traditional role to a more interconnected and cooperative relationship (under HCAI), the approach to measuring end-user characteristics evolves. The supporting figure (Figure 1) revisits the CAS investigation of Farooq and Grudin's example, illustrating how simple human interactions with computers are evolving into AI-integrated partnerships. The concept of interaction-to-integration misrepresentation highlights the shift in HCI from traditional user interfaces to a more collaborative HCAI approach.

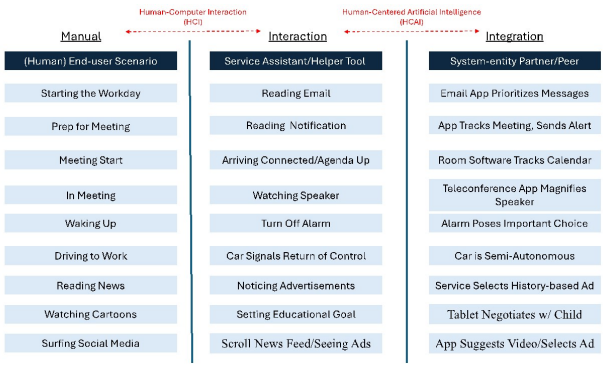

Figure 1: Interaction-to-integration use cases (Farooq and Grudin, 2016).

The figure visually maps this evolution by categorizing human activities, traditional service assistance and AI-enhanced system interactions, illustrating how HCI is shifting towards full integration.

Table 8: Interaction-to-integration usability misrepresentation.

Usability Factor	Misrepresentation Key Insight
Service Assistant to System Partner Transition	AI is no longer just a helper but a peer and collaborator, requiring autonomy and dynamic resource requests.
Situational & Contextual Awareness Challenges	The increasing complexity of automated service requests demands more sophisticated contextual understanding.
Mitigating Complexity w/ HCAI	The HCAI framework aims to correct usability misrepresentations in HCI, though it currently lacks comprehensive performance validation.
Security & Risk Considerations	The deeper integration of AI into decision-making and system automation raises concerns about security vulnerabilities.

Human-Centered HCI Framework Proposal

HCI primarily focuses on improving the usability, efficiency and effectiveness of interactive systems. The interaction is mostly command-based, where users request, then systems respond. HCAI involves AI systems that understand, learn, and collaborate with humans, anticipating needs and providing proactive support. The interaction becomes more context-aware and adaptive, integrating human factors deeply. Human-Centered HCI serves as a critical link in evolving HCI to HCAI, balancing human agency with AI-driven augmentation. Some initial observations to strategically bridge HCI to HCAI with Human-Centered HCI include the following as provided in Table 9.

Table 9: Human-centered HCI observations.

Strategy	Description	Example
Augmentation over Automation	Shift focus from replacing human effort to enhancing human capabilities.	AI-assisted medical diagnostics that support rather than replace physicians.
Contextual Adaptation	Develop AI that understands user context, environment, and preferences.	Virtual agents that adapt their responses based on a user's emotional state.
Ethical & Inclusive Design	Implement bias mitigation techniques to prevent discriminatory outcomes.	Inclusive AI systems that accommodate diverse accessibility needs.
Collaborative Systems	Create AI systems that act as collaborative partners, not just tools.	Co-creative AI platforms for design and innovation

By prioritizing transparency, ethics, trust, and collaboration, this approach not only enhances user experience but can also fosters societal acceptance of intelligent systems. Table 10 illustrates some core Human-Centered HCI principles established.

Table 10: Human-centered HCI principles.

Principle	Description	Application in HCAI
Transparency	Ensures AI decision-making is explainable and understandable.	Explaining AI decisions in healthcare for informed consent.
Trustworthiness	Builds confidence in AI systems through reliability and accountability.	Trustworthy autonomous vehicles adapting to unpredictable events.
Human Autonomy	Empowers users to maintain control over AI systems.	Adjustable AI in smart homes that respect user preferences.
Ethical Considerations	Aligns AI decisions with societal norms, ethics, and fairness.	Avoiding biased hiring algorithms in HR systems.
Context-Awareness	AI systems understand the context and adapt to human needs dynamically.	Context-aware assistants for personalized learning environments.

REFERENCES

Anderson, S. & Hartswood, M. & Procter, Rob & Rouncefield, M. & Slack, R. & Soutter, J. & Voss, A. (2003). Making autonomic computing systems accountable: The problem of human-computer interaction. 2003. 718–724. 10.1109/DEXA.2003.1232106.

Farooq, Umer and Grudin, Jonathan. (2016). Human-computer integration. interactions 23, 6 (November-December 2016), 26–32. https://doi.org/10.1145/3001896

Rufus, Rahmira S. (2023). "Complex Access Scenarios (CAS) Service Request." In HCI International 2023 – Late Breaking Posters (pp. 78–87). Springer, Cham.

Rufus, Rahmira S. (2021). "Intrusion Detection via Neuroception for an Autonomic Internet of Things." PhD diss., North Carolina Agricultural and Technical State University.